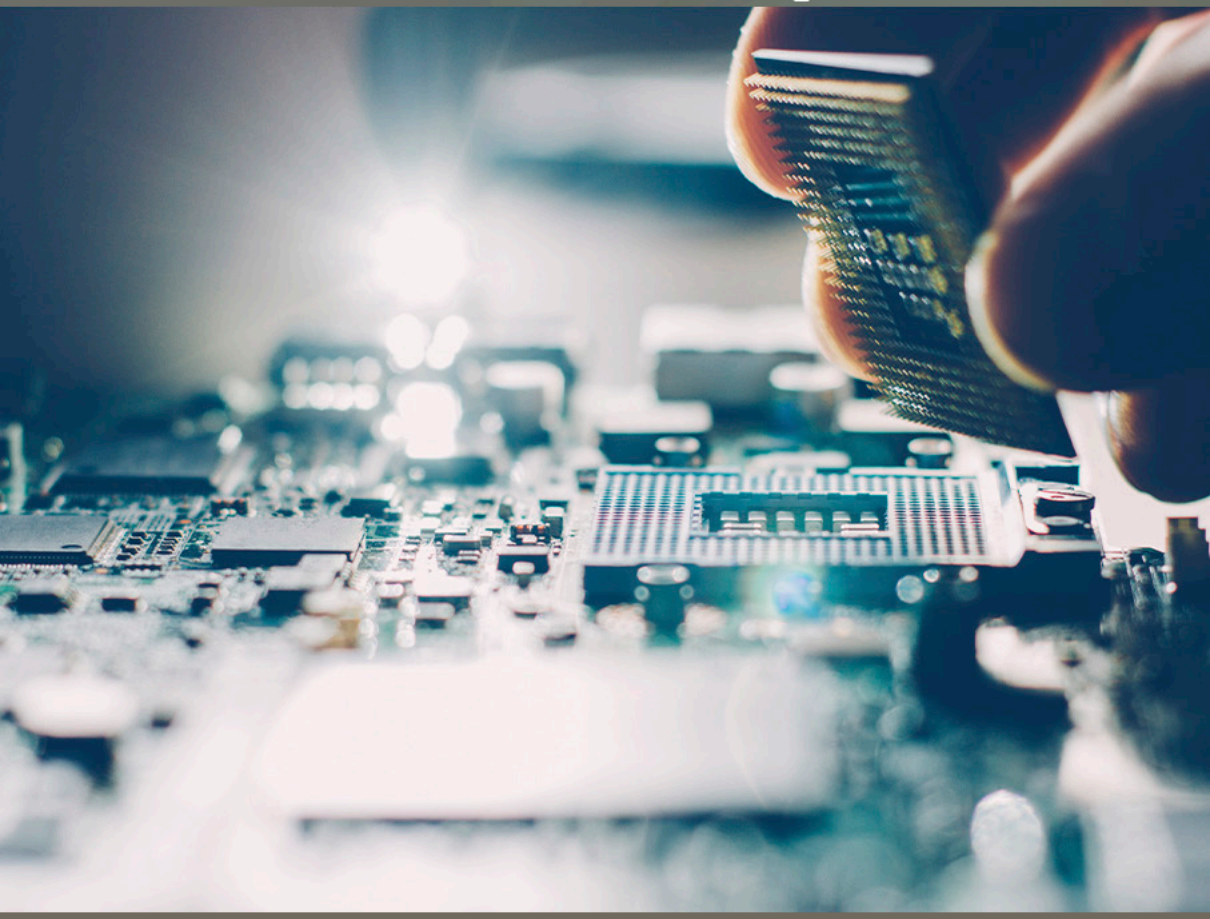
COLEÇÃO

DESAFIOS DAS ENGENHARIAS:

ENGENHARIA DE COMPUTAÇÃO



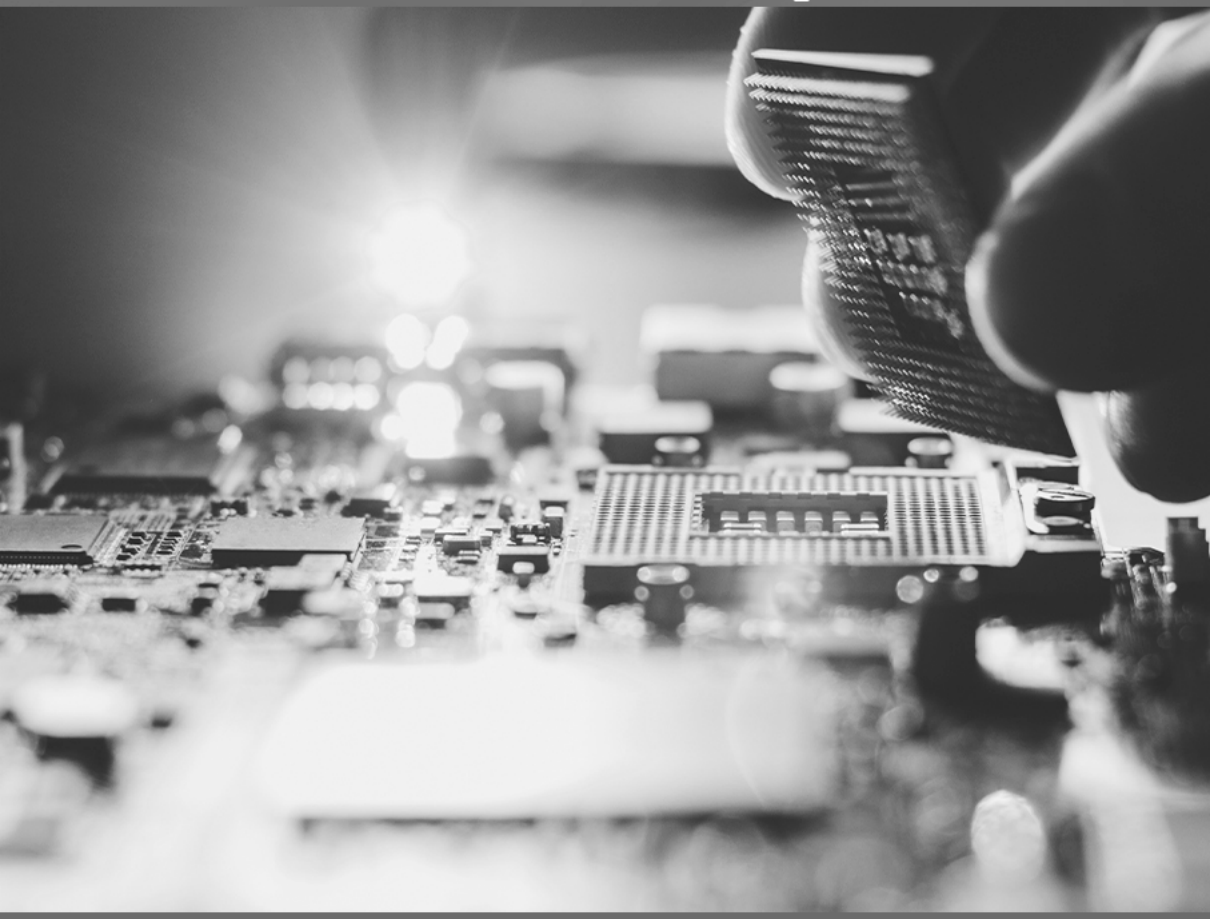
ERNANE ROSA MARTINS
(ORGANIZADOR)

Atena Editora
Ano 2021

COLEÇÃO

DESAFIOS DAS ENGENHARIAS:

ENGENHARIA DE COMPUTAÇÃO



ERNANE ROSA MARTINS
(ORGANIZADOR)

Atena Editora
Ano 2021

Editora chefe
Profª Drª Antonella Carvalho de Oliveira
Assistentes editoriais
Natalia Oliveira
Flávia Roberta Barão
Bibliotecária
Janaina Ramos
Projeto gráfico
Natália Sandrini de Azevedo
Camila Alves de Cremo
Luiza Alves Batista
Maria Alice Pinheiro
Imagens da capa
iStock
Edição de arte
Luiza Alves Batista
Revisão
Os autores

Atena Editora
Ano 2021

Ciências Agrárias e Multidisciplinar

Atena Editora
Ano 2021

Atena Editora
Ano 2021

Atena Editora
Ano 2021

Coleção desafios das engenharias: engenharia de computação

Diagramação: Camila Alves de Cremo
Correção: Maiara Ferreira
Indexação: Gabriel Motomu Teshima
Revisão: Os autores
Organizador: Ernane Rosa Martins

Dados Internacionais de Catalogação na Publicação (CIP)

C691 Coleção desafios das engenharias: engenharia de computação / Organizador Ernane Rosa Martins. – Ponta Grossa - PR: Atena, 2021.

Formato: PDF
Requisitos de sistema: Adobe Acrobat Reader
Modo de acesso: World Wide Web
Inclui bibliografia
ISBN 978-65-5983-387-0
DOI: https://doi.org/10.22533/at.ed.870211808

1. Engenharia da computação. I. Martins, Ernane Rosa (Organizador). II. Título.

CDD 621.39

Elaborado por Bibliotecária Janaina Ramos – CRB-8/9166

Atena Editora
Ponta Grossa – Paraná – Brasil
Telefone: +55 (42) 3323-5493
www.atenaeditora.com.br
contato@atenaeditora.com.br

Atena Editora
Ano 2021

Ciências Exatas e da Terra e Engenharias

DECLARAÇÃO DOS AUTORES

Os autores desta obra: 1. Atestam não possuir qualquer interesse comercial que constitua um conflito de interesses em relação ao artigo científico publicado; 2. Declaram que participaram ativamente da construção dos respectivos manuscritos, preferencialmente na: a) Concepção do estudo, e/ou aquisição de dados, e/ou análise e interpretação de dados; b) Elaboração do artigo ou revisão com vistas a tornar o material intelectualmente relevante; c) Aprovação final do manuscrito para submissão.; 3. Certificam que os artigos científicos publicados estão completamente isentos de dados e/ou resultados fraudulentos; 4. Confirmam a citação e a referência correta de todos os dados e de interpretações de dados de outras pesquisas; 5. Reconhecem terem informado todas as fontes de financiamento recebidas para a consecução da pesquisa; 6. Autorizam a edição da obra, que incluem os registros de ficha catalográfica, ISBN, DOI e demais indexadores, projeto visual e criação de capa, diagramação de miolo, assim como lançamento e divulgação da mesma conforme critérios da Atena Editora.

Atena Editora
Ano 2021

DECLARAÇÃO DA EDITORA

A Atena Editora declara, para os devidos fins de direito, que: 1. A presente publicação constitui apenas transferência temporária dos direitos autorais, direito sobre a publicação, inclusive não constitui responsabilidade solidária na criação dos manuscritos publicados, nos termos previstos na Lei sobre direitos autorais (Lei 9610/98), no art. 184 do Código penal e no art. 927 do Código Civil; 2. Autoriza e incentiva os autores a assinarem contratos com repositórios institucionais, com fins exclusivos de divulgação da obra, desde que com o devido reconhecimento de autoria e edição e sem qualquer finalidade comercial; 3. Todos os e-book são *open access, desta forma* não os comercializa em seu site, sites parceiros, plataformas de *e-commerce,* ou qualquer outro meio virtual ou físico, portanto, está isenta de repasses de direitos autorais aos autores; 4. Todos os membros do conselho editorial são doutores e vinculados a instituições de ensino superior públicas, conforme recomendação da CAPES para obtenção do Qualis livro; 5. Não cede, comercializa ou autoriza a utilização dos nomes e e-mails dos autores, bem como nenhum outro dado dos mesmos, para qualquer finalidade que não o escopo da divulgação desta obra.

Atena Editora
Ano 2021

APRESENTAÇÃO

A Engenharia de Computação tem como definição ser o ramo da engenharia que se caracteriza pelo projeto, desenvolvimento e implementação de sistemas, equipamentos e dispositivos computacionais, segundo uma visão integrada de hardware e software, apoiando-se em uma sólida base matemática e conhecimentos de fenômenos físicos. O objetivo é a aplicação das tecnologias de computação na solução de problemas de Engenharia.

Deste modo, este livro, aborda diversos aspectos tecnológicos computacionais, tais como: o desenvolvimento de um jogo de RPG acessível em LIBRAS; uma reflexão quanto à necessidade de aplicação de supressores de surto como proteção de transformadores devido a eventos transitórios em manobras de disjuntores; um algoritmo para geração de contorno 2D envolvendo regiões irregulares; avaliação da influência das tensões residuais e imperfeições geométricas iniciais em colunas de aço submetidas à flexão em torno do eixo de menor inércia; os esforços em estruturas laminares, de características de geometria e carregamentos diversos através da implementação computacional de um elemento finito sólido hexaédrico de 8 nós programado com uma linguagem computacional de alto nível; uma análise computacional realizada através do programa SAP2000; a estabilidade e as vibrações de anéis e tubulações apoiados em uma fundação elástica de Pasternak; um controlador neural para dois elos de um robô manipulador de três graus de liberdade (3 GDL); uma ferramenta de autoria para livros relacionados a área da educação; um aplicativo com propósito de aumentar a taxa de reciclagem e minimizar os danos ambientais devido ao descarte incorreto de resíduos na natureza; a conscientização de crianças e adolescentes sobre as ocorrências de bullying; uma aplicação web interativa, de fácil utilização e interface amigável, por meio do pacote Shiny, destinada aos tópicos de intervalo de confiança e dimensionamento de amostra para o parâmetro proporção; segmentar e detectar, por meio de redes neurais convolutivas, as pás dos raspadores de escória em panelas de ferro gusa do Reator Kambara de uma siderúrgica; integrar a Biblioteca Digital de Artigos (IFPublica) e a Plataforma de Digital de Inscrição e Administração de Projetos (PDIAP), por meio de adaptações nos dois projetos, para impedir erros humanos e automatizar o processo de cadastro de artigos do PDIAP na base de dados do IFPublica.

Assim, espero que a presente obra venha a se tornar um guia aos estudantes e profissionais da área de Engenharia de Computação, auxiliando-os em diversos assuntos relevantes da área, fornecendo a estes novos conhecimentos para poderem atender as necessidades informacionais, computacionais e de automação das organizações de uma forma geral. Por fim, agradeço aos autores por suas contribuições na construção desta importante obra e desejo muito sucesso a todos os nossos leitores.

Ernane Rosa Martins

SUMÁRIO

CAPÍTULO 26

FLUID FLOW SUMMARIZATION USING DYNAMIC MULTI-VECTOR FEATURE SPACES

Data de aceite: 02/08/2021
Data de submissão: 29/04/2021

Renato José Policani Borseti
National Laboratory for Scientific Computing
Petropolis – RJ – Brazil
http://lattes.cnpq.br/4272517293880873

Leandro Tavares da Silva
Federal Center of Technology Education Celso Suckow da Fonseca
Petropolis – RJ – Brazil
http://lattes.cnpq.br/1045203549961256

Gilson Antonio Giraldi
National Laboratory for Scientific Computing
Petropolis – RJ – Brazil
http://lattes.cnpq.br/9950879952262717

ABSTRACT: Recent works have applied summarization concepts for fluid flow analysis in computational fluid dynamics (CFD) simulations to yield a synthetic and useful visual abstraction of the flow evolution. In one of these works, the pipeline firstly performs a coarse temporal stream flow segmentation that is automatically improved by k-means to complete the visual summary. The original technique considers, as the input data, only the velocity field and the particles configuration obtained by smoothed particle hydrodynamics (SPH) simulations. In this work, we demonstrate a relationship between the vorticity field and the pressure gradient. The obtained result points toward the necessity of combining the vorticity and pressure gradient as well as the velocity and particles configuration fields to compose the feature space to search for the fundamental segments of the fluid evolution. Besides, we incorporate an interval tree to improve computation of the coarse flow segmentation. We demonstrate the methodology using a 2D SPH simulation of the N-roll mill apparatus where N= 6 symmetrically placed rollers, that rotate at constant angular velocities, are surrounded by a fluid. We show that the clusters generated by our algorithm captures a compact but detailed picture of important segments of the fluid.

KEYWORDS: Computational Fluid Dynamics, Summarization, Smoothed Particle Hydrodynamics, N-Roll Mill Flow, Dynamic Vectors.

SUMARIZAÇÃO DE FLUXO DE FLUIDO USANDO ESPAÇOS DINÂMICOS CARACTERÍSTICA DE MULTI-VETOR

RESUMO: Trabalhos recentes têm aplicado conceitos de sumarização para análise de fluxo de fluidos em simulações de dinâmica de fluidos computacional (CFD) para produzir uma abstração visual sintética e útil da evolução do fluxo. Em um desses trabalhos, a tubulação executa primeiramente uma segmentação temporal grosseira do fluxo de fluido que é automaticamente melhorada por meios k para completar o resumo visual. A técnica original considera, como dados de entrada, apenas o campo de velocidade e a configuração das partículas obtidas por simulações de hidrodinâmica de partículas suavizadas (SPH). Neste trabalho, demonstramos uma relação entre o campo de vorticidade e o gradiente de pressão. O resultado obtido aponta para a necessidade

de combinar o campo de vorticidade e o gradiente de pressão, assim como os campos de velocidade e configuração de partículas para compor o espaço de características para a busca dos segmentos fundamentais da evolução do fluido. Além disso, incorporamos uma árvore de intervalos para melhorar o cálculo da segmentação do fluxo grosseiro. Demonstramos a metodologia utilizando uma simulação 2D SPH do aparelho do moinho de rolos N onde N = 6 rolos simetricamente colocados, que giram a velocidades angulares constantes, são circundados por um fluido. Mostramos que os clusters gerados por nosso algoritmo capturam um quadro compacto, mas detalhado, de segmentos importantes do fluido.
PALAVRAS-CHAVE: Dinâmica de fluídos computacional, sumarização, SPH, fluxo N-roll, dinâmica vetorial.

1 | INTRODUCTION

The process of data analysis in computational fluid dynamics (CFD) is computationally intensive, being a problem well-suited to scientific visualization and visual data mining. The techniques in scientific visualization convert numerical data into visual representations meaningful for scientific analysis Rosemblum et al. (1994). Visual data mining refers to works that use graphical mapping and interaction techniques for visual representation and analysis of databases de Oliveira & Levkowitz (2003); Simoff et al. (2008). In the specific case of CFD data, the main challenges in both cases include to mine, track, and visualize the important features in the data. In this area, researchers aim the development of unified frameworks to detect the significant features as well as to follow and explain them in the time-varying fields (tracking process) Yang et al. (2005).

Both scientific visualization and visual data mining methods can be included in problem solving environments (PSE), with the goal of integration of a problem domain oriented specification language with CFD solvers, data analysis, and component software technology. From a more practical viewpoint, the goal is to implement a computational environment that offers to the scientists a broad range of tools for modeling, simulating, visualizing, and iteratively exploring a problem space Houstis & Rice (2000). That is the point where fluid flow summarization touches PSE fundamental proposals. For instance, CFD simulations undergo issues about boundary effects, numerical and physical parameters that intrude upon the numerical simulation results. Therefore, over time, an increasing set of solutions can be obtained with an increasing range of setups targeting multiple instances of the same phenomenon, while largely retaining the way problems are expressed. Consequently, fluidicists need a PSE that incorporates specific metrics and visualization techniques for comparing raw data produced by numerical simulations. In the case of time dependent numerical fields, data analysis and comparison procedures should process the temporal data sequence for information extraction. Such task resembles computer vision issues in video processing applications.

A video stream is a temporal series that refers to a sequence of successive happenings captured by a camera over a time interval. The video structure is generally described in a

descending hierarchy that starts with video clips, followed by scenes, shots, and ending with frames. Video processing encompasses representation, video compression, segmentation, enhancement, video analysis as well as video summarization Bovik (2005). The latter is a class of algorithms that aims to generate a summary of the whole video sequence. The obtained video summary must give a synthetic visual representation of the video information Elkhattabi et al. (2015).

The key idea of the work presented in Silva & Giraldi (2016) is to apply the concepts behind video summarization for fluid flow analysis in CFD simulations. The goal is to build a synthetic visual representation of the time evolution of the fluid, based on the detection of fundamental sequences in the fluid flow, also called shots. The methodology focus on macro-scale models for fluid description which can be derived through mass and momentum conservation laws in conjunction with Smoothed Particle Hydrodynamics (SPH), that is a general meshfree technique used to simulate a widely range of natural phenomena Monaghan (2012). The fluid summarization technique described in Silva & Giraldi (2016) considers, as the input data, only the velocity field and the particles configuration obtained by SPH simulations.

In this paper, we firstly demonstrate the relationship between the vorticity field and the pressure gradient. The obtained result points toward the necessity of combining the vorticity and pressure gradient, besides the velocity and particles configuration fields used in Silva & Giraldi (2016), to compose the feature space to search for the fundamental segments of the fluid evolution. Hence, following Silva & Giraldi (2016), we firstly perform a coarse temporal stream flow segmentation. The obtained result is used to define the initial centroids for k-means which automatically improves the initial segments in the fluid flow. For each obtained cluster, a keyframe is selected through a refined procedure. However, differently from Silva & Giraldi (2016), the course flow segmentation is yielded through an interval tree which allows to simplify the parameter choice and improve the obtained result.

We demonstrate the methodology using numerical results obtained from a 2D fluid simulation performed by SPH. We take the SPH implementation of the N-roll mill presented in Silva & Giraldi (2015). The N-roll mill set up consists of N rollers which are surrounded by a fluid. The rollers are symmetrically placed and they rotate with constant angular velocities. Also, the fluid domain is a rectangular region $U \subset \mathbb{R}^2$ and, consequently, we add suitable boundary conditions. We simulate the N-roll mill flows, for $N = 6$ using the SPH technique. The obtained result is used to generate summaries using several combinations of the vector quantities involved. The obtained summaries are compared using a temporal pyramid representation of the flow and statistical procedure, also developed in Silva & Giraldi (2015).

We show that, in the performed tests, the clusters generated by our algorithm when adding the vorticity and pressure gradient to the velocity/particle fields capture a detailed picture of important segments of the fluid.

The paper is organized as follows. In section 2 we survey works related with our

proposal. Section 3 shows the relationship between the vorticity and the pressure gradient. Section 4 describes the SPH method and its application for N -roll mill simulation. In section 5 we described the proposed summarization method. The methodology for summaries evaluation is reviewed in section 6. The computational experiments are presented on section 7. Conclusions and future directions for this work are given on section 8.

2 | RELATED WORK

Our proposal for fluid flow summarization shares the main philosophy of shot-based video summarization techniques: perform a temporal segmentation of the data stream in order to find out the fundamental shots in the frame sequence. Most of the video summarization techniques work in feature spaces to build a synthetic and useful visual abstraction of video sequences Elkhattabi et al. (2015). The key idea behind feature extraction is the representation of a pattern using a higher level description. This reduces the amount of data and provides a crucial first step for video analysis in general.

According to Elkhattabi et al. (2015), the summary may either be static, if it is composed by representative frames (keyframes), or dynamic, if the summarization consists in moving images extracted from the original video (video skimming). The process of retaining only the essential information of a video sequence (or CFD simulation) improves analysis, retrieval, bandwidth, and navigation inside large data archives.

Up to the best of our knowledge, the references Wang et al. (2008); Silva & Giraldi (2016) are the only works that applies summarization techniques for CFD data analysis in a systematic procedure. In the former authors introduce an importance-driven approach to time-varying volume data visualization for enhancing the ability to extract the most essential aspects of timevarying data. The methodology is based on a block-wise analysis of the data that derives an importance curve for each data block based on the formulation of conditional entropy. Drawing the importance curve for the whole volume, by applying the methodology to compute the importance values for each time step of the simulation, allows to summarizes the overall temporal behavior of the time-varying data. On the other hand, in Silva & Giraldi (2016), the central idea of the whole technique is the concept of shots, imported from video processing field. In this proposal, a shot is a fundamental sequence in the fluid flow. Authors define similarity measures to compare distinct frames of the CFD simulation and apply the k-means algorithm to compute the clusters that generates the shots. Firstly, a coarse stream flow segmentation is performed using a suitable similarity measure. The obtained result is used to define the initial centroids for the machine learning setup. The k-means result automatically improves the initial segments of the fluid flow and consequently allows to extract the correct shots from the original flow evolution. For each obtained cluster a keyframe is selected by user interaction or through a refined procedure. In this way, it is incorporated machine learning and video summarization techniques for

fluid flow analysis. The result is a static summary because it is composed by representative keyframes.

The just cited techniques falls into the class of visualization methods for CFD analysis which can be divided into scientific visualization and visual data mining groups. The techniques in scientific visualization can be classified according to the data type they manage. Scalar fields ($F : D \subset \mathbb{R}^3 \rightarrow \mathbb{R}$), vector fields ($F(x)$ is a vector, $x \in D \subset \mathbb{R}^3$) and a tensor fields compose the usual range of data types in this area. Henceforth, we have methods for scalar fields visualization (isosurface generation, volume rendering, colormap, etc.), vector fields visualization (field lines computation, particle tracing, topology of vector fields, line integral convolution (LIC), among others) and techniques for tensor fields (topology and hyperstreamlines) Rosemblum et al. (1994).

Visual data mining refers to works that use graphical mapping and interaction techniques for visual representation and analysis of databases de Oliveira & Levkowitz (2003); Simoff et al. (2008). In the specific case of CFD data, the main challenges include to mine, track, and visualize the important features in the data. In this area, researchers aim the development of unified frameworks to detect the significant features as well as to follow and explain them in the time-varying fields (tracking process) Yang et al. (2005).

3 | NAVIER-STOKES AND ROTATING SYSTEMS

In the Eulerian formulation of fluid mechanics, the governing equations of the fluid which connect the pressure p, density ρ and velocity $\mathbf{v} = (v_1, v_2, v_3)$ are the continuity (mass conservation) and the Navier-Stokes equations. The former is given by:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0. \tag{1}$$

We consider the Navier-Stokes equation in the form Hirsch (1988):

$$\rho \frac{d\mathbf{v}}{dt} = -\nabla p + \nu \Delta \mathbf{v} + \rho \mathbf{F}, \tag{2}$$

where *dv*/*dt* is the material derivative:

$$\frac{d\mathbf{v}}{dt} = \frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v}. \tag{3}$$

In the case of incompressible fluids we must add the constraint $\vec{\nabla} \cdot \vec{v} = 0$. Besides, we need a state equation for the pressure field:

$$p = c^2 \rho \tag{4}$$

where *c* is the speed of sound.

Besides the fields mentioned above, the curl of the velocity, named vorticity, is

another fundamental quantity for fluid motion understanding Hirsch (1988). This fact motivates works in vortex visualization that apply the following heuristic: vortex are regions composed by points with high vorticity and low pressure Garth et al. (2004). In this section we present a mathematical analysis to justify such heuristic. So, let two reference systems ($\mathbf{u}_1$, $\mathbf{u}_2$, $\mathbf{u}_3$) and $\left(\overset{*}{\mathbf{u}}_1, \overset{*}{\mathbf{u}}_2, \overset{*}{\mathbf{u}}_3\right)$ with common origin, where the system $*$ is not fixed. So, the time-dependent position vector $\mathbf{r} = x_1\mathbf{u}_1 + x_2\mathbf{u}_2 + x_3\mathbf{u}_3$ can be represented in both reference systems as: $\mathbf{r} = \sum_{i=1}^{3} x_i\mathbf{u}_i = \sum_{i=1}^{3} x_i^* \overset{*}{\mathbf{u}}_i$.

If we supposed a time-dependent vector **r** = **r**(*t*), we obtain the following expressions for the time derivative in both the reference systems:

$$\frac{d^*\mathbf{r}}{dt} = \sum_{i=1}^{3} \frac{dx_i^*}{dt} \overset{*}{\mathbf{u}}_i, \tag{5}$$

$$\frac{d\mathbf{r}}{dt} = \sum_{i=1}^{3} \frac{dx_i^*}{dt} \overset{*}{\mathbf{u}}_i + \sum_{i=1}^{3} x_i^* \frac{d\overset{*}{\mathbf{u}}_i}{dt}. \tag{6}$$

Now, we suppose that $\overset{*}{\mathbf{u}}_3$ is constant and the reference system $*$ rotates with constant angular velocity given by $\mathbf{\Omega} = \beta\overset{*}{\mathbf{u}}_3$, where β is a constant also. Then, we can show that:

$$\frac{d\overset{*}{\mathbf{u}}_i}{dt} = \mathbf{\Omega} \times \overset{*}{\mathbf{u}}_i, \ i = 1, 2, 3. \tag{7}$$

Therefore, by substituting expressions (5) and (7) into equation (6) we get:

$$\frac{d\mathbf{r}}{dt} = \frac{d^*\mathbf{r}}{dt} + \mathbf{\Omega} \times \mathbf{r}. \tag{8}$$

Now, we analyze the continuity equation in rotating reference system considering the mass density in the system $*$ given by $\rho^* = \rho^*\left(x_1^*, x_2^*, x_3^*, t\right)$. So, the continuity equation (1) in the system $*$ has the form:

$$\frac{\partial\rho^*}{\partial t} + \nabla^*\left(\rho^*\mathbf{v}^*\right) = 0, \tag{9}$$

where $\mathbf{v}^* = \frac{d^*\mathbf{r}}{dt}$.

On the other hand, the derivative of expression (8) respect to time gives:

$$\frac{d^2\mathbf{r}}{dt^2} = \frac{d^{*2}\mathbf{r}}{dt^2} + 2\mathbf{\Omega} \times \frac{d^*\mathbf{r}}{dt} + \mathbf{\Omega} \times (\mathbf{\Omega} \times \mathbf{r}) + \frac{d\mathbf{\Omega}}{dt} \times \mathbf{r}. \tag{10}$$

The Navier-Stokes equation, given by expressions (2)-(3), in the reference system $*$ can be written by replacing the external field **F*** by the acceleration $\frac{d^{*2}\mathbf{r}}{dt^2}$:

$$\rho^* \frac{\partial}{\partial t}(\mathbf{v}^*) + \mathbf{v}^* \cdot \nabla \mathbf{v}^* = \rho^* \frac{d^{*2}\mathbf{r}}{dt^2} - \nabla p^* + \nu \Delta \mathbf{v}^*. \quad (11)$$

We can obtain $\frac{d^{*2}\mathbf{r}}{dt^2}$ through equation (10), with the simplification that $\frac{d\mathbf{\Omega}}{dt} = \mathbf{0}$, due to the fact that Ω = *constant*. Consequently, expression (11), becomes:

$$\rho^* \frac{\partial}{\partial t}(\mathbf{v}^*) + \mathbf{v}^* \cdot \nabla \mathbf{v}^* = \rho^* \mathbf{F} - \nabla p^* + \nu \Delta \mathbf{v}^* - 2\rho^* \mathbf{\Omega} \times \mathbf{v}^* - \rho^* \mathbf{\Omega} \times (\mathbf{\Omega} \times \mathbf{r}), \quad (12)$$

where use replace $\frac{d^{*2}\mathbf{r}}{dt^2}$ by the external field **F**. If v* = 0, the equation (12) simplifies to:

$$\nabla^* \cdot p^* = \rho^* \mathbf{F} - \rho^* \mathbf{\Omega} \times (\mathbf{\Omega} \times \mathbf{r}), \quad (13)$$

where the second term in the right hand side is the centripetal acceleration. Thus, regardless of the external field **F**, we see from expression (13) that the pressure gradient in the reference system $*$ points towards the exterior of the vortex, which justifies the idea that vortex points have low pressure. Since vorticity is related to the angular velocity Saffman (1992), it is also justified the idea that the points of the vortex must have high vorticity.

4 | SPH VERSION OF FLUID EQUATIONS

The two fundamental elements in the SPH method are an interpolation kernel W : $\mathbb{R}^3 \rightarrow \mathbb{R}^+$, which is a symmetric function respect to the origin (0, 0, 0), limited, with compact support, and a particle system $\mathbf{r}_i = (x_{i1}, x_{i2}, x_{i3}) \in \mathbb{R}^3$, $i = 1, 2, \ldots, M$, that represents a discrete version (sample) of the fluid. The kernel estimate of a scalar quantity *A* and its gradient in a point $\mathbf{r}_i \in \mathbb{R}^3$ are given by Liu & Liu (2003):

$$\langle A(\mathbf{r}_i)\rangle = \sum_{j=1}^{M} \frac{m_j}{\rho(\mathbf{r}_j)} A(\mathbf{r}_j) W(\mathbf{r}_i - \mathbf{r}_j, h), \quad \langle \nabla A(\mathbf{r}_i)\rangle = \sum_{j=1}^{M} \frac{m_j}{\rho(\mathbf{r}_j)} A(\mathbf{r}_j) \nabla_i W(\mathbf{r}_i - \mathbf{r}_j, h), \quad (14)$$

where $\nabla_i W(\mathbf{r}_i - \mathbf{r}_j, h)$ means $\nabla_{\mathbf{r}} W(\mathbf{r} - \mathbf{r}_j, h)$ evaluated at $\mathbf{r} = \mathbf{r}_i$, h is the smoothing length which determines the support of the kernel, and $\rho(\mathbf{r}_j)$ is the density at the particle position $\mathbf{r}_j$ Liu & Liu (2003). Therefore, the kernel estimate of the density at the position $\mathbf{r}_i$ is:

$$\langle \rho(\mathbf{r}_i)\rangle = \sum_{j=1}^{M} m_j W(\mathbf{r}_i - \mathbf{r}_j, h), \quad (15)$$

Besides, we can show that the divergent for a vector field v can be computed as Liu & Liu (2003):

$$\langle \nabla \cdot \mathbf{v}(\mathbf{r}_i)\rangle = \sum_{j=1}^{M} \frac{m_j}{\rho(\mathbf{r}_j)} (\mathbf{v}(\mathbf{r}_j) - \mathbf{v}(\mathbf{r}_i)) \nabla_i W(\mathbf{r}_i - \mathbf{r}_j, h), \quad (16)$$

An analogous expression can be obtained by the Laplacian. From equations (14)

and (16) we can observe that there is no need for a mesh to compute spatial derivatives. With equations (14)-(16), we are ready to write the discrete version of the fluid equations of section 3. For simplicity, we will remove the brackets and place a subscript i on the denoted value; that is: $\langle A(\mathbf{r}_i)\rangle \equiv A_i$ and $\langle \nabla A(\mathbf{r}_i)\rangle \equiv \nabla A_i$. We also use the notations $W_{ij} \equiv W(\mathbf{r}_i - \mathbf{r}_j, h)$ and $\rho(\mathbf{r}_j) \equiv \rho_j$. By using the material derivative for the density $(d\rho/dt = \partial\rho/\partial t + \nabla\rho \cdot \mathbf{v})$ as well as expressions (14)-(16) we can show that the SPH version of the time derivative of the density becomes:

$$\frac{d\rho_i}{dt} = \sum_{j=1}^{M} m_j \mathbf{v}_{ij} \nabla_i W_{ij}, \tag{17}$$

where $\mathbf{v}_{ij} \equiv \mathbf{v}_i - \mathbf{v}_j$.

Now, we shall obtain the discrete version of the Navier-Stokes equation given by expression (2). Firstly, the acceleration that particle i feels due to the pressure force only is given by Liu & Liu (2003):

$$\frac{d\mathbf{v}_i}{dt} = -\sum_{j=1}^{M} m_j \left(\frac{p_j}{\rho_j^2} + \frac{p_i}{\rho_i^2}\right) \nabla_i W_{ij}. \tag{18}$$

Due to numerical stability issues we follow Liu & Liu (2003) and use the artificial viscosity computed by:

$$\Pi^{ij} = \begin{cases} \dfrac{-2(a\varphi_{ij}c + b\varphi_{ij}^2)}{\rho^i + \rho^j}, & \mathbf{v}^{ij}\cdot\mathbf{x}^{ij} < 0 \\ 0, & \mathbf{v}^{ij}\cdot\mathbf{x}^{ij} \geq 0 \end{cases} \tag{19}$$

with $\varphi_{ij} = (\mathbf{v}^{ij}\cdot\mathbf{x}^{ij})h/\left(\left(r_k^{ij}\right)^2 + 0.01h^2\right)$, where *a*, *b* are parameters, $\mathbf{v}^{ij} = \mathbf{v}^i - \mathbf{v}^j$, $\mathbf{x}^{ij} = \mathbf{r}^i - \mathbf{r}^j$.

Actual viscous effects can be introduced in the SPH framework through a kernel version of the viscous term in equation (2), given by:

$$\nabla^2\mathbf{v}_i = 2.0\frac{\nu}{\rho_i}\sum_{j=1}^{N}\frac{m_j\mathbf{v}^{ij}}{\rho_j}\frac{\mathbf{x}^{ij}\cdot\nabla_i W_{ij}}{r^{ij}}, \tag{20}$$

where $\mathbf{v}^{ij}$, $\mathbf{x}^{ij}$, r^{ij}, are defined above.

Besides, boundary particles that do not move but interact with fluid particles are included to implement repulsive boundary forces to prevent the interior particle from penetrating the boundary of the domain Liu & Liu (2003). If a boundary particle q^g is the neighboring particle of a real particle q^i_k that is approaching the boundary, then the force:

$$\Gamma_k^{ig} = \begin{cases} D\left[\left(\frac{r_0}{r_k^{ig}}\right)^{n_1} - \left(\frac{r_0}{r_k^{ig}}\right)^{n_2}\right] & \frac{r_0}{r_k^{ig}} \leq 1 \\ 0 & \frac{r_0}{r_k^{ig}} > 1 \end{cases} \tag{21}$$

is applied pairwise along the centerline of these two particles, where $n_1 = 12$, $n_2 = 4$, $r_k^{ig} = \|\mathbf{x}^i - \mathbf{x}^g\|$, r_0 is usually selected close to the initial particle spacing, and *D* is problem dependent parameter. Once the force terms are computed, the acceleration of particle *i* can be obtained by:

$$\mathbf{a}_i = \frac{d\mathbf{v}_i}{dt} = -\frac{\nabla p_i}{\rho_i} + \frac{1}{\rho_i}\left(\mu\nabla^2\mathbf{v}_i + \Pi^{ij} + \Gamma^{ig}\right) + \mathbf{F}_i. \tag{22}$$

Thus, an integration scheme, the Leap-Frog one, is used in order to update the velocity and the positions of particles as follows:

$$\mathbf{v}_i^{t+\Delta t} = \mathbf{v}_i^t + \frac{\Delta t}{2}\mathbf{a}_i^t, \quad \mathbf{r}_i^{t+\Delta t} = \mathbf{r}_i^t + \Delta t\mathbf{v}_i^{t+\Delta t} \tag{23}$$

where Δt is the time step for the integrator.

In this work, the kernel function adopted is the quintic smoothing function:

$$W(R) = \begin{cases} (3-R)^5 - 6(2-R)^5 - 15(1-R)^5, & 0 \leq R < 1 \\ (3-R)^5 - 6(2-R)^5, & 1 \leq R < 2 \\ (3-R)^5, & 2 \leq R < 3 \\ 0, & R \geq 3 \end{cases} \tag{24}$$

where $R = \frac{\|\mathbf{r}_i - \mathbf{r}_j\|}{h}$, and *h* is a constant.

In Silva & Giraldi (2015), the SPH framework to simulate the *n*-roll mill setup uses two types of particles: SPH particles and roll particles. The former is used to simulate the fluid while the latter are uniformly distributed along the rolls boundary and they simulate the interaction between rolls and the fluid. In the simulation pipeline, the SPH particle are processed by: (a) Compute the pressure field using expression (4) and the density at previous time; (b) Update the density field by integrating equation (17); (c) Compute the accelerations by expression (22); (d) Update the velocity and the position of SPH particles through equations (23). In our implementation rolls particles are treated like the SPH ones. However, we do not update position and velocity of rolls particles. Consequently, we only update the pressure and density at their positions in each integration step. This simple strategy simulates the idea of rigid rolls whose physical properties are not affected due to the fluid surrounding them.

5 | PROPOSED METHODOLOGY

Our proposal involves the analysis of two-dimensional flows simulated in a domain $D \subset \mathbb{R}^2$ through the SPH technique described on section 4. Each time step of the simulation updates the particles positions and computes the pressure and velocity fields. With these quantities, we can derive other dynamic fields, like vorticity and pressure gradient Liu & Liu (2003). Anyway, we shall notice that each of these quantities (except the pressure) can be represented as vectors in the two-dimensional Euclidean space, denoted by E^2. So, generically speaking, let $\mathcal{X}$ a dynamic SPH field:

$$\mathcal{X} = \left\{ (\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_M) \in E^2 \times E^2 \times \cdots \times E^2 \right\}, \tag{25}$$

where $\mathbf{x}_i = \mathbf{x}(\mathbf{r}_i) \in \mathbb{R}^2$ with $\mathbf{r}_i$, $i = 1, 2, \cdots, M$, being the position of SPH particles.

We need specific metrics to compute the similarity between two frames, say $\mathbf{z}, \mathbf{w} \in \mathcal{X}$ of the simulation. In this paper, we apply the metrics $d_{\mathcal{X}} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^+$ and $d_{\mathcal{X}max} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^+$ computed, respectively, by:

$$d_{\mathcal{X}}(\mathbf{z}, \mathbf{w}) = \left[\sum_{i=1}^{M} \left((z_{i1} - w_{i1})^2 + (z_{i2} - w_{i2})^2 \right) \right]^{1/2}, \tag{26}$$

$$d_{\mathcal{X}max}(\mathbf{z}, \mathbf{w}) = max \left\{ \left((z_{i1} - w_{i1})^2 + (z_{i2} - w_{i2})^2 \right)^{1/2}, i = 1, 2, \cdots, M \right\}, \tag{27}$$

to perform this task Silva & Giraldi (2016).

An important point of our proposal is to combine different fields to build feature spaces that are input to generate flow summaries. So, if we have ϱ SPH dynamical fields $\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_\varrho$ (for instance, position, velocity, vorticity and pressure gradient), we can build the multi-vector space $\mathcal{F} = \mathcal{X}_1 \times \mathcal{X}_2 \times \ldots \times \mathcal{X}_\varrho$ and generalize expressions (26)-(27) to this space. However, such metrics will involve different physical fields with quantities that could vary in different scales and ranges. Consequently, the variable that is spread out over a larger scale/range may dominate the other ones which implies that we can not decide whether a given distance value is significant or not.

So, we need some normalization procedure once our approach is steered by multivariate data analysis techniques. Henceforth, let us suppose a generic SPH field $\mathcal{X}$ and *N* steps of SPH simulation, which generates a data set *S* composed by the sequence of numerical frames $(\mathbf{x}(t_j)) \in \mathcal{X}$, $j = 0, 1, 2, \cdots, N$, where:

$$\mathbf{x}(t_j) = \left(\mathbf{x}_1^j, \mathbf{x}_2^j, \cdots, \mathbf{x}_M^j \right) \equiv \mathbf{x}^j, \quad j = 0, 1, 2, \cdots, N, \tag{28}$$

with $\mathbf{x}_i^j = \left(x_{i,1}^j, x_{i,2}^j \right) \in E^2$ being $\mathbf{x}_i^j = \mathbf{x}\left(\mathbf{r}_i^j \right)$ where $\mathbf{r}_i^j$ means position of the particle i at simulation time $t = t_j$. A solution to address the normalization requirement is to center each frame x^j respect to the corresponding data mean and, next, to perform spatial normalization by using the standard deviations for avoiding that variables with larger scales

(or variances) dominate. Formally, we replace the Euclidean distance (26) by a simplified version of the Mahalanobis distance, computed by the following procedure:

1. Compute the means:

$$\overline{\mathbf{x}} = \frac{1}{N}\sum_{j=0}^{N}\mathbf{x}^j, \tag{29}$$

2. Centering data:

$$\widetilde{\mathbf{x}}^j = \mathbf{x}^j - \overline{\mathbf{x}} = \left(x_{11}^j, \widetilde{x}_{12}^j; \widetilde{x}_{21}^j, \widetilde{x}_{22}^j; \widetilde{x}_{31}^j, \widetilde{x}_{32}^j; \cdots; \widetilde{x}_{M1}^j, \widetilde{x}_{M2}^j\right), \tag{30}$$

for $j = 0, 1, 2, \cdots, N$.

3. Compute scatter vectors:

$$\widetilde{\Psi}_{ik} = \left(\frac{1}{N}\sum_{j=1}^{N}|\widetilde{x}_{ik}^j|^2\right)^{1/2},\ k = 1, 2;\ i = 1, 2, \cdots, M, \tag{31}$$

4. Normalization, generating the space $\widehat{\mathcal{X}}$:

$$\hat{\mathbf{x}}^j = \left(\frac{\widetilde{x}_{11}^j}{\widetilde{\Psi}_{11}}, \frac{\widetilde{x}_{12}^j}{\widetilde{\Psi}_{12}}; \frac{\widetilde{x}_{21}^j}{\widetilde{\Psi}_{21}}, \frac{\widetilde{x}_{22}^j}{\widetilde{\Psi}_{22}}; \cdots; \frac{\widetilde{x}_{M1}^j}{\widetilde{\Psi}_{M1}}, \frac{\widetilde{x}_{M2}^j}{\widetilde{\Psi}_{M2}}\right), \tag{32}$$

With the result of *N* steps of SPH simulation, we can build a sequence $(\mathbf{x}(t_j)) \in \mathcal{F}$, $j = 0, 1, 2, \cdots, N$, in the $\mathcal{F} = \mathcal{X}_1 \times \mathcal{X}_2 \times \ldots \times \mathcal{X}_\varrho$ space, where:

$$\mathbf{x}(t_j) = \left(\mathbf{x}_1^j, \mathbf{x}_2^j, \cdots, \mathbf{x}_M^j\right) \equiv \mathbf{x}^j, \quad j = 0, 1, 2, \cdots, N, \tag{33}$$

with $\mathbf{x}_i^j = \left(\mathbf{x}_{i,1}^j, \mathbf{x}_{i,2}^j, \cdots, \mathbf{x}_{i,M}\right) \in \mathcal{X}_i$, $i = 1, 2, \ldots, \varrho$. So, we perform the normalization of each space $\mathcal{X}_i$, following the above procedure, to get the normalized spaces $\widehat{\mathcal{X}}_1, \widehat{\mathcal{X}}_2, \ldots, \widehat{\mathcal{X}}_\varrho$, and define the multi-vector feature space $\widehat{\mathcal{F}} = \widehat{\mathcal{X}}_1 \times \widehat{\mathcal{X}}_2 \times \ldots \times \widehat{\mathcal{X}}_\varrho$ for summary computation. Hence, we can compute the distance between two points $\hat{\mathbf{x}} = (\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2, \ldots, \hat{\mathbf{x}}_\varrho) \in \widehat{\mathcal{F}}$ and $\hat{\mathbf{y}} = (\hat{\mathbf{y}}_1, \hat{\mathbf{y}}_2, \ldots, \hat{\mathbf{y}}_\varrho) \in \widehat{\mathcal{F}}$ using expressions analogous to (26) and (27) as follows:

$$d_{EF} : \widehat{\mathcal{F}} \times \widehat{\mathcal{F}} \to \mathbb{R}^+, \quad d_{EF}(\hat{\mathbf{x}}, \hat{\mathbf{y}}, \alpha) = \left[\sum_{i=1}^{\varrho}\alpha_i d_{\mathcal{X}}^2(\hat{\mathbf{x}}, \hat{\mathbf{y}})\right]^{1/2}, \tag{34}$$

$$d_{Fmax} : \widehat{\mathcal{F}} \times \widehat{\mathcal{F}} \to \mathbb{R}^+, \quad d_{Fmax}(\hat{\mathbf{x}}, \hat{\mathbf{y}}) = max\left\{d_{\mathcal{X}}(\hat{\mathbf{x}}_i, \hat{\mathbf{y}}_i), \quad i = 1, 2, \ldots, \varrho\right\}. \tag{35}$$

where $\alpha = (\alpha_i, \alpha_2, \ldots, \alpha_\varrho) \in \{0, 1\}^\varrho$ and $d_{\mathcal{X}}^2(\hat{\mathbf{x}}_i, \hat{\mathbf{y}}_i) = \left(d_{\mathcal{X}}(\hat{\mathbf{x}}_i, \hat{\mathbf{y}}_i)\right)^2$ with $d_{\mathcal{X}}$ given by expression (26).

As already stated before, the first step of our methodology is to yield a coarse temporal flow segmentation. We claim that, if the feature space $\widehat{\mathcal{F}}$ incorporates enough information about the physical system then the expressions (34)-(35) can be used as similarity measures to summarize the flow evolution. In this way, let a subsequence $L(a, b) = (r(a), r(a+1), \ldots r(b))$, where $r(a) = d(\widehat{\mathbf{x}}^a, \widehat{\mathbf{x}}^{a+1})$, $a = 0, 1, 2, \cdots, N-1$,

with $d : \widehat{\mathcal{F}} \times \widehat{\mathcal{F}} \rightarrow \mathbb{R}^{+}$ denoting the metric (35) or (34). Let also the minimum and maximum values of the list denoted by max (L(*a,b*)) and min (L(*a,b* respectively, and a threshold δ which is used as the stop criterion *C*:

$$C\left(a,b\right) = \begin{cases} 1, & if \quad |\max\left(L\left(a,b\right)\right) - \min\left(L\left(a,b\right)\right)| < \delta, \\ 0, & otherwise, \end{cases} \tag{36}$$

Now, we build an interval tree to subdivide the list *L* (0, *N* - 1) into homogeneous parts according to the criterion 36. Hence, the level zero of the tree holds the whole list: *a* = 0 e *b* = *N*-1. If the test in expression (36) returns 0, then the algorithm builds the level 1, as indicated in the Figure 1. In this step, the expression (36) is evaluated for both the lists *L* (1, (*N* - 1)/2) and *L* ((*N* -1)/2 + 1, *N*-1) . They are recursively divided in the same manner before until the test in expression (36) returns true or there are no sublists left. Obviously, we can perform an analogous process if we apply the metric (27) or (26) to build the list *L* (0, *N* - 1) .

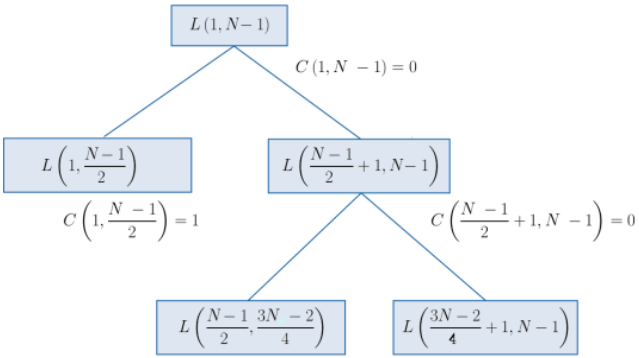


Figure 1: Interval tree construction.

This step yields a set of initial segments (intervals) of the simulation. We take the last frame in each segment, generating a list $\overline{\mathbf{q}}^{i,0}$, $i = 1, 2, \cdots, k$, with *k* being the number of leafs in the interval tree and $\overline{\mathbf{q}}^{i,0} \in \mathcal{X}$ or $\overline{\mathbf{q}}^{i,0} \in \widehat{\mathcal{F}}$, depending on the space we are working. In the second step, we apply our k-means implementation, initialized with the centroids being the frames selected in the first stage, to complete the clustering process (see Jain et al. (1999) for k-means details). The distance between two frames in the k-means algorithm, in the space $\widehat{\mathcal{F}}$, is obtained by expression (34), with α = (1, 1,..., 1). For each obtained cluster, a keyframe is selected through a suitable metric. The proposed method can be summarized by the Algorithm 1. To simplify the explanation, we use a generic variable '**q**' to represent a SPH field in the $\mathcal{X}$ or $\mathcal{F}$ spaces. In the Algorithm 1, the procedure *Interval – Tree* implements the interval tree computation, with subsequent choice of the initial centroids, according to the explanation given above. Next, the *K – means* computation is performed,

using a straightforward adaptation of the traditional procedure (see Silva & Giraldi (2016) for details).

6 | SUMMARY EVALUATION

Algorithm 1 Algorithm for compute a summary of the SPH simulation.

SPH simulation data set $S = \{\mathbf{q}^j,\ j = 0, 1, 2, \cdots, N\}$; Thresholds δ, ε;
if $S \subset \mathcal{F}$ **then**
  Data normalization through expressions (29)-(32);
  $d_1 \leftarrow$ metric in expression (35);
  $d_2 \leftarrow$ metric in expression (34), with $\alpha = (1, 1, \ldots, 1)$;
  $d_3 \leftarrow$ metric in expression (34), with $\alpha = (1, 1, \ldots, 1)$;
else
  $d_1 \leftarrow$ metric in expression (27);
  $d_2 \leftarrow$ metric in expression (26);
  $d_3 \leftarrow$ metric in expression (26);
end if
Compute the initial centroids: $\left(\overline{\mathbf{q}}^{i,0}\right)_{i=1,2,\cdots,k} = Interval - Tree\left(d_1, S, \delta\right)$;
Solve k-means: $\left(S_i, \left(\overline{\mathbf{q}}^i\right)\right)_{i=1,2,\cdots,k} = K - means\left(d_2, \varepsilon, S, \left(\overline{\mathbf{q}}^{i,0}\right)_{i=1,2,\cdots,k}\right)$;
Solve the minimization problem:

$$\overline{\overline{\mathbf{q}}}^i = \arg \min_{\mathbf{q}^j \in S_i} d_3\left(\mathbf{q}^j, \overline{\mathbf{q}}^i\right), \tag{37}$$

for $i = 1, 2, \cdots, k$ to find the keyframes
Output: Summary, composed by the keyframe sequence:

$$\overline{\overline{\mathbf{q}}}^1, \overline{\overline{\mathbf{q}}}^2, \cdots, \overline{\overline{\mathbf{q}}}^k.$$

Now, we must evaluate the obtained summary. To perform this task, we follow Silva & Giraldi (2016), and use a similarity measure based on the metrics given in section 5 and an evaluation methodology based on a temporal pyramid representation and sampling in the time domain of the simulation. Specifically, the summary generated by the Algorithm 1 can be represented by $SU = \left\{\left(S_i, \overline{\overline{\mathbf{q}}}^i\right),\ i = 1, 2, \cdots, k\right\}$, where each S_I is a cluster and $\overline{\overline{\mathbf{q}}}^i$ the corresponding keyframe, calculated through equation (37) in Algorithm 1. Therefore, we can compute the similarity between the original simulation S and the summary *SU* through the function:

$$f\left(S, SU\right) = \frac{1}{N} \sum_{i=1}^{k} \sum_{\mathbf{q}^j \in S_i} d\left(\mathbf{q}^j, \overline{\overline{\mathbf{q}}}^i\right), \tag{38}$$

where *d* denotes the metric (26), if S $\subset \mathcal{X}$ or the metric (34) if S $\subset \mathcal{F}$.

We can notice that this expression gives the average of the distances between the keyframes sequence and the entire simulation. Therefore, it makes sense to compute the corresponding variance and standard deviation given, respectively, by:

$$Var\left(S,SU\right)=\frac{1}{N}\sum_{i=1}^{k}\sum_{\mathbf{q}^j\in S_i}\left|d\left(\mathbf{q}^j,\overline{\mathbf{q}}^i\right)-f\left(S,SU\right)\right|^2, \tag{39}$$

$$Std\left(S,SU\right)=Var\left(S,SU\right)^{1/2}, \tag{40}$$

We cover two aspects in the remainder of this section. Firstly, the comparison between a summary and the entire simulation result. Second, the comparison among two distinct summaries obtained by our methodology. The pyramid model applied in the tests of this section takes the numerical frames $\mathbf{q}^j$, j = 0, 1, 2, ... , N, and keeps only the even ones to build level 1. The level 2 is composed by the frames of the original simulation with index i = 0, 4, 8,... . Therefore, the level s will be formed by the frames indexed originally by $i = 2^s_j$, j = 0, 1, 2,...[$N/2^s$] . The obtained structure is represented on Figure 2 and the corresponding temporal sequences are denoted by: Ls = {$\mathbf{q}^i$; $i=2^sj$, *with* j = 0,1,2,...[$N/2^s$]} .

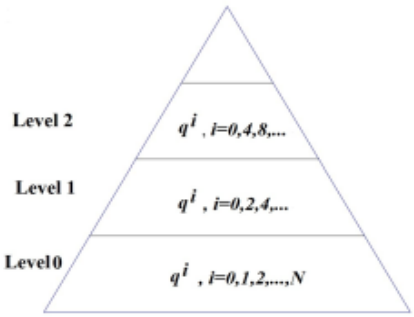


Figure 2: Pyramid for summary evaluation.

The level 1 of the obtained pyramid can be considered as a 'good' summary in the sense that it is representative of the original video. The next levels gradually lose the representative capability. If expression (38) is an efficient measure to compare two distinct frame sequences then we expect that the distances from a good summary to the levels zero and one are small. However, it is also expected that $f(SU, L_s)$ increases when going up the pyramid structure ($s > 1$), because this indicates that the summaries encompasses important features of the simulation which are lost in the higher levels of the pyramid. That is why we expect that the similarity decreases for L_s with $s > 1$ if SU is a good summary.

However, we should be careful to use the statistics given by expressions (38)-(40) for comparisons between the summaries built using different spaces. In fact, we need a relative quantity to be able to compare the summaries. A possibility to perform this task is the transformation:

$$\Omega_{SU} = \frac{f(L_0, SU) + Std(L_0, SU)}{f(L_0, SU)} = 1 + \frac{Std(L_0, SU)}{f(L_0, SU)}, \tag{41}$$

which is a measure of the uncertainty in the efficiency of a summary *SU* compared to the value of the similarity measure given by expression (38), with $S = L_0$.

7 | COMPUTATIONAL EXPERIMENTS

In this section we test the summarization scheme computed by Algorithm 1. In these experiments we consider the following spaces: (i) Particles positions, or configuration space (*C*); (ii) The velocity space (*V*); (iii) Vorticity field (Ω); (iv) Pressure gradient (∇_p); (v) Normalized multi-vector feature spaces; (v.1) Configuration and velocity ($\widehat{\mathcal{F}}_1 = \widehat{\mathcal{C}}\widehat{\mathcal{V}}$); (v.2) Vorticity and pressure gradient ($\widehat{\mathcal{F}}_2 = \widehat{\Omega}\widehat{\nabla}_p$); (v.3) Velocity and vorticity spaces ($\widehat{\mathcal{F}}_3 = \hat{\mathcal{V}}\widehat{\Omega}$); (v.4) Velocity, vorticity, and pressure gradient ($\widehat{\mathcal{F}}_4 = \hat{\mathcal{V}}\widehat{\Omega}\widehat{\nabla}_p$); (v.5) Configuration, velocity, vorticity, and pressure gradient ($\widehat{\mathcal{F}}_5 = \widehat{\mathcal{C}}\hat{\mathcal{V}}\widehat{\Omega}\widehat{\nabla}_p$).

We test the methodology using a bidimensional fluid simulations of *N* -roll mill setup with *N* = 6, in a domain $U \subset \mathbb{R}^2$, filled by a viscous fluid, as pictured in Figure 3. The flow visualization is implemented using the line integral convolution technique (LIC), applied over the velocity field. The LIC is available in the Paraview system Henderson & Ahrens (2004), adapted for visualization of SPH results. All the simulations are performed using the SPH approach described on section 4.

For all cases, the kernel function is calculated by equation (24) with smoothing length $h = 0.006$. The rest density is $\rho_0 = 1261.3\ kg/m^3$, $a = b = 0.9$ in the artificial viscosity, and the simulation domain *U* is filled by a fluid with the glycerol viscosity ($\mu = 1.5\ Pa \cdot s^{-1}$). The 6-roll mill apparatus, represented on Figure 3, is simulated using a computational domain with dimensions $D_x = 0.38$, $D_y = 0.36\ m$ and the roller centers are symmetrically placed in the points $C_1 = (0.24, 0.1794)$, $C_2 = (0.2155, 0.223)$, $C_3 = (0.1655, 0.2236)$, and $C_4 = (0.14, 0.1806)$, $C_5 = (0.1645, 0.137)$, and $C_6 = (0.2145, 0.137)$. They have radius $a = 0.01\ m$ and separation $d = 0.04\ m$. We consider three angular velocities for the rolls, say $\Omega_3 = \Omega_5 = 200.8740$, $\Omega_1 = \Omega_4 = 276.5262$ e $\Omega_2 = \Omega_6 = 293.3413$ The simulation uses $M = 5131$ SPH particles, each one with mass $0.03286kg$. At the initialization, the particles are uniformly distributed along the simulation domain *U*, being initialized with null velocities.

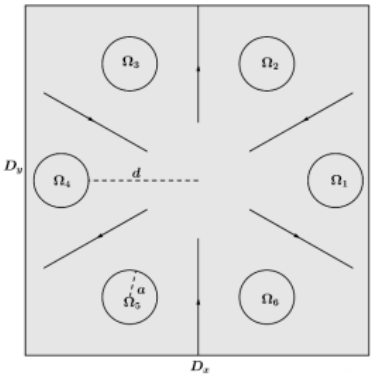


Figure 3: The 6-roll mill apparatus scheme.

We need an approach that allows to analyze the flow patterns transitions along the time evolution of the simulation. To address this issue we plot the points $(i,\ d(\mathbf{q}^i,\ \mathbf{q}^{i+1}))$, i=0,1,2,..., N, to obtain the Figure 4, where d is the metric (26), if $S \subset \mathcal{X}$, or the metric (34) if $S \subset \mathcal{F}$. The plots in Figure 4 present in the horizontal axis the time index 'i' and in the vertical axis the corresponding distances.

From the Figures 4.(a)-(d) we can identify 4 stages in the flow: (1) From t = 0*seg* to t = 0.003*seg*; (2) Between t = 0.0031 and t = 0.006; (3)Between 0.0061 and 0.02 (4) After t= 0.021. The Figures 5.(a)-(l) show some snapshots of the fluid motion, generated through LIC, taken in the mentioned intervals[1].

1 Complete frame sequence in http://virtual01.lncc.br/~giraldi/simulations/6RollMill/video6rollmill.avi

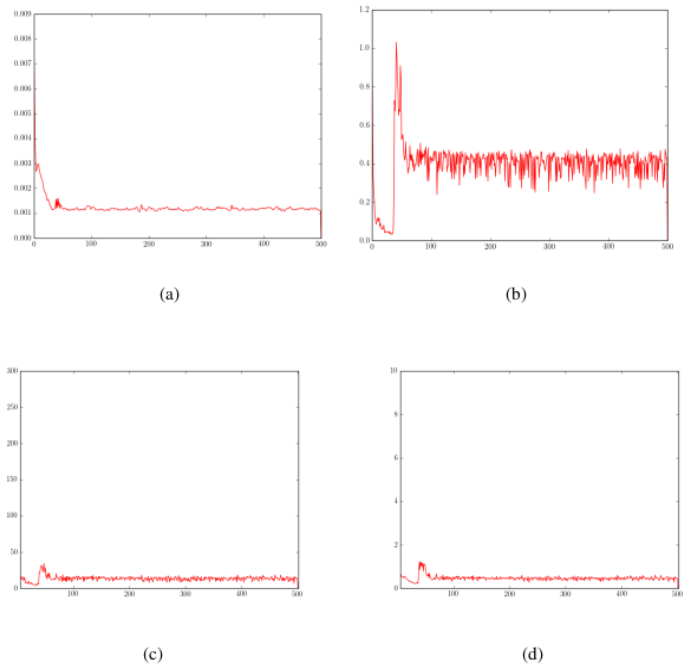


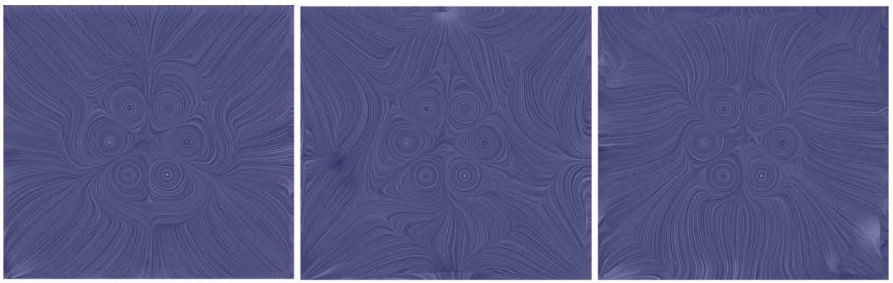
Figure 4: (a) Time evolution of the distance $d_{\mathcal{X}}\left(\mathbf{q}^i, \mathbf{q}^{i+1}\right)$, wich $\mathbf{q}^i \in \mathcal{C}$. (b) Analogous to the previous figure but using the distance $d_{\mathcal{X}}\left(\mathbf{q}^i, \mathbf{q}^{i+1}\right)$, with $\mathbf{q}^i \in \mathcal{V}$. (c) Alalogous but with $\mathbf{q}^i \in \Omega\nabla_p$ and $d_{EF}\left(\mathbf{q}^i, \mathbf{q}^{i+1}, (1,1)\right)$. (d) Plot for the distances $d_{EF}\left(\mathbf{q}^i, \mathbf{q}^{i+1}, (1,1,1,1)\right)$ with $\mathbf{q}^i \in \mathcal{CV}\Omega\nabla_p$.

The first issue to yield the summary using the Algorithm 1 is to set the parameter δ used to build the interval tree (equation (36)). The Figures 4.(a)-(d) indicate a kind of 'quasi' stead state configuration for $t > 200$ which can be also observed from Figures 5.(j)-(l). We must choose the parameter δ in order to avoid splitting too much the interval [200, 500] as well as to capture important pattern changes. So, in what follows we test two values:

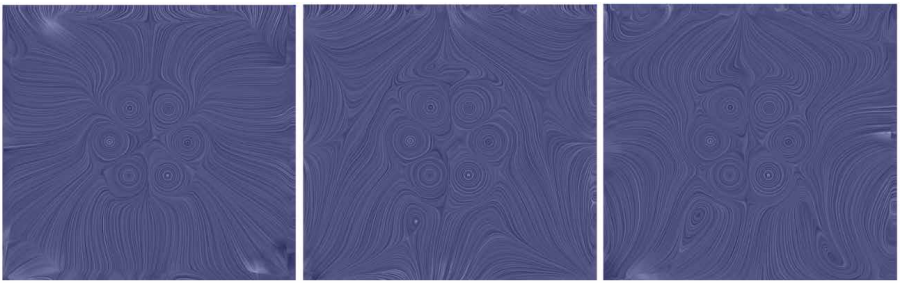
$$\delta_1 = |\max\left(L\left(300, 500\right)\right) - \min\left(L\left(300, 500\right)\right)|, \tag{42}$$

and $\delta_2 = \delta_1 /2$. The choice between these two values is made through trial-and-error method steered by the guesses that it is difficult that few numerical frames could suitably represent the simulation and, on the other hand, too many numerical frames will contain redundancies.

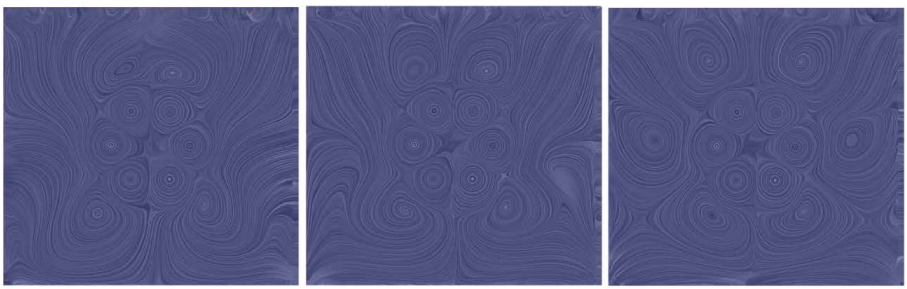
Hence, the Table 1 helps to analyze the sensitivity of the Algorithm 1 respect to the threshold δ. For instance, we notice that the number of frames of the summary in the $\widehat{\mathcal{F}}_5 = \widehat{\mathcal{C}}\widehat{\mathcal{V}}\widehat{\Omega}\widehat{\nabla}_p$ space increases from 12 to 28 when changing the threshold from $\delta = \delta_1$ to $\delta = \delta_1 /2$.



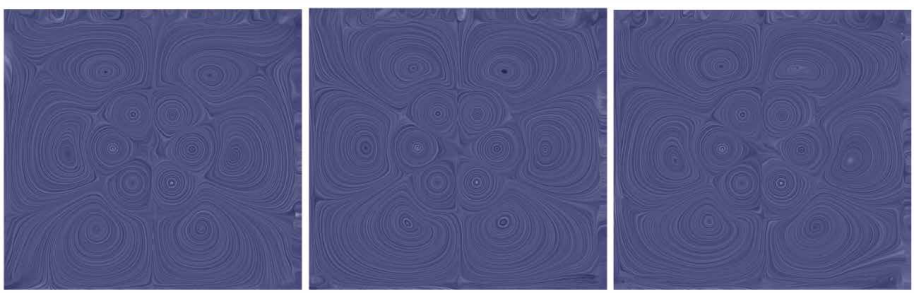
(a) $j = 15$ (b) $j = 20$ (c) $j = 30$



(d) $j = 32$ (e) $j = 41$ (f) $j = 50$



(g) $j = 55$ (h) $j = 60$ (i) $j = 75$



(j) $j = 100$ (k) $j = 200$ (l) $j = 450$

Figure 5: Six roll mill configuration at time steps $t+j \cdot \Delta t$.

	$\mathcal{C}$	$\mathcal{V}$	Ω	∇_p	$\widehat{\mathcal{F}}_1$	$\widehat{\mathcal{F}}_2$	$\widehat{\mathcal{F}}_3$	$\widehat{\mathcal{F}}_4$	$\widehat{\mathcal{F}}_5$
$\delta = \delta_1/2$	21	29	38	33	35	37	38	28	28
$\delta = \delta_1$	10	15	13	22	21	17	16	12	12

Table 1: Sensitivity of Algorithm 1 to δ value in expression (36) using the vector and multi-vector spaces listed in the first line. The second line reports the number of frames in the summary obtained by using for $\delta = \delta_1$, where δ_1 is computed by expression (42). Analogously for the third line but using $\delta = \delta_1/2$. As already defined, we use the notations: $\widehat{\mathcal{F}}_1 = \widehat{\mathcal{C}}\widehat{\mathcal{V}}$, $\widehat{\mathcal{F}}_2 = \widehat{\Omega}\widehat{\nabla}_p$, $\widehat{\mathcal{F}}_3 = \widehat{\mathcal{V}}\widehat{\Omega}$, $\widehat{\mathcal{F}}_4 = \widehat{\mathcal{V}}\widehat{\Omega}\widehat{\nabla}_p$, $\widehat{\mathcal{F}}_5 = \widehat{\mathcal{C}}\widehat{\mathcal{V}}\widehat{\Omega}\widehat{\nabla}_p$.

However, taking into account that the whole simulation has 500 frames, we consider that the number of keyframes reported in Table 1 is not too large. So, we go ahead with the tests and evaluate all the summaries through the methodology described on section 6. The Table 2 shows the f values, as well as the corresponding standard deviations, computed by expressions (38)-(40) when using the metric the metric (26), if $S \subset \mathcal{X}$, with $\mathcal{X} \in C, V, \Omega, \nabla_p$ or the metric (34) if S is one of the multi-vector spaces.

	$\mathcal{C}: f$	$\mathcal{C}: Std$	$\mathcal{V}: f$	$\mathcal{V}: Std$	$\Omega: f$	$\Omega: Std$	$\nabla_p: f$	$\nabla_p: Std$
L_0	0.09	0.05	1.11	0.31	64.02	52.05	205.25	92.79
L_1	0.28	0.19	3.33	2.24	192.24	138.34	616.36	421.39
L_2	0.66	0.56	7.75	6.64	444.79	384.18	1434.21	1232.23
L_3	1.39	1.30	16.59	15.48	953.12	890.68	3075.29	2871.22
L_4	2.83	2.74	33.75	32.64	1941.40	1878.22	6249.41	6044.87

(a)

	$\widehat{\mathcal{F}}_1: f$	$\widehat{\mathcal{F}}_1: Std$	$\widehat{\mathcal{F}}_2: f$	$\widehat{\mathcal{F}}_2: Std$	$\widehat{\mathcal{F}}_3: f$	$\widehat{\mathcal{F}}_3: Std$	$\widehat{\mathcal{F}}_4: f$	$\widehat{\mathcal{F}}_4: Std$	$\widehat{\mathcal{F}}_5: f$	$\widehat{\mathcal{F}}_5: Std$
L_0	0.07	0.04	60.25	20.87	2.13	0.72	7.69	2.73	7.70	2.73
L_1	0.22	0.15	180.67	122.25	6.39	4.32	23.06	15.61	23.11	15.65
L_2	0.51	0.44	419.88	360.30	14.86	12.75	53.57	45.97	53.72	46.10
L_3	1.09	1.02	899.46	839.57	31.84	29.72	114.72	107.09	115.19	107.52
L_4	2.22	2.14	1827.75	1767.83	64.83	62.70	234.07	226.40	234.63	226.96

(b)

Table 2: Similarity *f* and standard deviation *Std*, computed by expressions (38) and (40), using the metric (26), for N-roll mill simulation with $N = 6$, for $\delta = \delta_1/2$. (b) Analogous result, but in the multivector spaces, using the metric (34) to calculate equations (38) and (40).

We notice that all the obtained summaries follow the expected behavior, reported in the last paragraph of section 6: smaller values for $f(L_0, SU)$ and $f(L_1, SU)$, with $f(L_0, SU) \leq f(L_1, SU) \leq f(L_2, SU) \leq ... \leq f(L_4, SU)$. This result indicates that the summaries encompasses

important features of the simulation which are lost in the higher levels of the pyramid. That is why the similarity decreases for L_s with $s > 1$. The same behavior is observed for the summaries obtained with $\delta = \delta_1$ in the interval tree.

Now, we apply the expression (41) to compare the summaries. The expression (41) gives the results reported on Table 3 for all the summaries.

	$\mathcal{C}$	$\mathcal{V}$	Ω	∇_p	$\widehat{\mathcal{F}}_1$	$\widehat{\mathcal{F}}_2$	$\widehat{\mathcal{F}}_3$	$\widehat{\mathcal{F}}_4$	$\widehat{\mathcal{F}}_5$
$Std/f : \delta = \delta_1/2$	0.57	0.28	0.81	0.45	0.57	0.35	0.34	0.36	0.35
$Std/f : \delta = \delta_1$	0.48	0.26	0.54	0.36	0.55	0.24	0.26	0.23	0.23

Table 3: Relative interval computed by expression (41) for the summaries SU generated in all the considered spaces.

The results reported in Table 3 indicate a superiority of the summaries obtained in the multi-vector spaces $\widehat{\mathcal{F}}_2$, $\widehat{\mathcal{F}}_4$, and $\widehat{\mathcal{F}}_5$, when setting for $\delta = \delta_1$, respect to the relative error Ω_{SU} . The Table 4.(a) reports the keyframes of the corresponding summaries while Table 4.(b) shows the number of keyframes in each interval identified in Figure 4.(a)-(d).

	Keyframes	δ
$\widehat{\mathcal{F}}_2$	$\{0, 2, 8, 11, 14, 19, 26, 35, 47, 64, 85, 112, 186, 211, 315, 400, 414\}$	δ_1
$\widehat{\mathcal{F}}_4$	$\{0, 4, 9, 18, 34, 53, 241, 266, 293, 319, 352, 383\}$	δ_1
$\widehat{\mathcal{F}}_5$	$\{0, 5, 14, 31, 51, 74, 268, 294, 319, 352, 383, 413\}$	δ_1
Manual Summary	$\{15, 20, 30, 32, 41, 50, 55, 60, 75, 100, 200, 450\}$	–

(a)

Intervals	$0-30$	$31-60$	$61-200$	$201-500$
$\widehat{\mathcal{F}}_2$:No Keyframes	7	2	4	4
$\widehat{\mathcal{F}}_4$:No Keyframes	4	2	0	6
$\widehat{\mathcal{F}}_5$:No Keyframes	3	2	1	6
Manual Summary:No Keyframes	3	5	3	1

(b)

Table 4: (a) Keyframe sequences for the best and manual summaries with the corresponding multi-vector spaces and threshold δ. (b) Number of keyframes in the important intervals, for the best and manual summaries.

We notice that the important stages of the fluid evolution are covered by the keyframes, except in the case of $\widehat{\mathcal{F}}_4$, space that has zero keyframes in the interval 61 – 200. The summary in $\widehat{\mathcal{F}}_5$ space (with $\delta = \delta_1$) covers all the important intervals. It has only six keyframe (two more than the $\widehat{\mathcal{F}}_2$) because in the last interval, for $i > 200$, the flow achieves a 'quasi' stead state configuration, with few significant changes along the time, as

we can confirm in the Figures 5.(k)-(l) of the manual summary. Besides, the other keyframes cover important flow patterns in the first intervals. Such observations agree with the fact that the relative interval of the summary in $\widehat{\mathcal{F}}_5$ is small or equal the other ones. Also, we shall highlight that the two best summaries incorporate the vorticity and pressure gradient spaces, which agrees with our claim that these quantities should be incorporated in the methodology presented in Silva & Giraldi (2016).

In order to compare the above result and the original methodology Silva & Giraldi (2016) we shall highlight that, in the original initial flow segmentation depends on the parameter T = $\mathcal{T}\triangle$ where $\triangle = (1/\bar{N})\sum_{i=1}^{N} d\left(\mathbf{q}^i, \mathbf{q}^0\right)$ and $\mathcal{T}$ is chosen through trial and error tests. In situations like the 6 – *roll* mill experiment that has a stead state configuration with non null velocity field, the determination of $\mathcal{T}$ is more difficult than the determination of δ because, like performed in the above experiments, to set δ it is just a matter of localize the stead state region in the distance plot and compute expression (42). Also, the work Silva & Giraldi (2016) only considers the configuration and velocity spaces that do not offer the best summaries, according to Table 3.

8 | CONCLUSIONS AND FUTURE WORKS

In this work, we the technique proposed in Silva & Giraldi (2016) by considering dynamic multi-vector spaces and incorporating an interval tree to improve the determination of the initial shots. We extract summaries from a SPH simulations of the 6–*roll* mill setup of Figures 3. The quality of the best summary was confirmed by the quantitative methodology and the comparison with the manual summary. We must improve the methodology to deal with situations where two summaries get the same value for the quantity *Std/f* , as happening in Table 3. Also, we must compare our method with the reference Wang et al. (2008).

REFERENCES

Bovik, Alan C. 2005. **Handbook of Image and Video Processing (Communications, Networking and Multimedia).** Orlando, FL, USA: Academic Press, Inc.

de Oliveira, Maria Cristina Ferreira, & Levkowitz, Haim. 2003. **From Visual Data Exploration to Visual Data Mining: A Survey**. IEEE Trans. Vis. Comput. Graph., 9(3), 378–394.

Elkhattabi, Zaynab, Tabii, Youness, & Benkaddour, Abdelhamid. 2015. **Video Summarization: Techniques and Applications**. Int. Journal of Comp., Elect., Autom., Control and Inf. Eng., 9(4), 809 – 814.

Garth, Christoph, Tricoche, Xavier, Salzbrunn, Tobias, Bobach, Tom, & Scheuermann, Gerik. 2004. **Surface Techniques for Vortex Visualization**. Pages 155–164 of: Proc. of the Sixth Joint Eurographics - IEEE TCVG Conference on Visualization. VISSYM'04.

Henderson, Amy, & Ahrens, Jim. 2004. **The Paraview guide : a parallel visualization application**. New York: Kitware, Inc.

Hirsch, C. 1988. **Numerical Computation of Internal and External Flows**. Vol. 1. Livros Tcnicos e Cientficos Editora S.A., RJ-Brasil.

Houstis, Elias N., & Rice, John R. 2000. **Future problem solving environments for computational science**. Mathematics and Computers in Simulation Journal, 54(4–5), 243–257.

Jain, A. K., Murty, M. N., & Flynn, P. J. 1999. **Data Clustering: A Review**. ACM Comput. Surv., 31(3), 264–323.

Liu, G.R., & Liu, B. 2003. **Smoothed Particle Hydrodynamics: A Meshfree Particle Method**. World Scientific.

Monaghan, J. J. 2012. **Smoothed Particle Hydrodynamics and Its Diverse Applications**. Annual Review of Fluid Mechanics, 44(1), 323–346.

Rosemblum, L., Earnshaw, R., Encarnacao, J., Hagen, H., Kaufman, A., Klimenko, S.V., Nielson, G., Post, F., & Thalmann, D. 1994. **Scientific Visualization: Advances and Challenges**. Academic Press.

Saffman, P. G. 1992. **Vortex Dynamics**. Cambridge University Press.

Silva, L.T., & Giraldi, G. 2015. **Visualizing Morse Flows and Elliptic Umbilic Catastrophes in N-Roll Mill Fluids Simulated by Smoothed Particle Hydrodynamics**. Pages 1–20 of: XXXVI Ibero-Latin American Congress on Computational Methods in Engineering - CILAMCE. Rio de Janeiro, RJ.

Silva, L.T., & Giraldi, G. 2016. **Fluid Flow Summarization Technique for Computational Fluid Dynamics Analysis**. Pages 1–44 of: Computational Fluid Dynamics (CFD): Characteristics, Applications and Analysis. Nova Science Publishers. Chapter 3.

Simoff, Simeon, Bohlen, Michael H., & Mazeika, Arturas (eds). 2008. In: Simoff, Simeon, Bohlen, Michael H., & Mazeika, Arturas (eds), **Visual Data Mining: Theory, Techniques and Tools for Visual Analytics**. Berlin, Heidelberg, New York: Springer.

Wang, C., Yu, H., & Ma, K. L. 2008. **Importance-Driven Time-Varying Data Visualization**. IEEE Trans. on Vis. and Comp. Graphics, 14(6), 1547–1554.

Yang, Hui, Parthasarathy, Srinivasan, & Mehta, Sameep. 2005. **A Generalized Framework for Mining Spatio-temporal Patterns in Scientific Data**. Pages 716–721 of: Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining. New York, NY, USA: ACM.

ÍNDICE REMISSIVO

W

COLEÇÃO

DESAFIOS DAS ENGENHARIAS:

ENGENHARIA DE COMPUTAÇÃO

www.atenaeditora.com.br
contato@atenaeditora.com.br
@atenaeditora
www.facebook.com/atenaeditora.com.br

Atena Editora
Ano 2021

COLEÇÃO

DESAFIOS DAS ENGENHARIAS:

ENGENHARIA DE COMPUTAÇÃO

www.atenaeditora.com.br
contato@atenaeditora.com.br
@atenaeditora
www.facebook.com/atenaeditora.com.br

Atena Editora
Ano 2021